Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

Vehicle Extraction from Aerial Images Using Voting Process and Frame Matching

Kensuke Tanaka and Hitoshi Saji

Abstract— **In today's motorized society, traffic congestion and illegal parking are serious problems in cities. We must collect traffic information to solve these problems. Fixed cameras or sensors set up on roads have been used so far. However, approaches using such cameras have many disadvantages. For example, these cameras cannot be used in the event of accident, and it is difficult to collect information for a wide area at one time. On the other hand, a survey of areas by many people requires much time and money. There is a real need to simplify these approaches. In our study, we extract vehicles from a wide area of roads using aerial images, and we propose a method for helping to obtain traffic information.**

I. INTRODUCTION

Vehicles have become popular now and play a significant role in our lives. However, the increasing number of vehicles causes problems of traffic congestion, illegal parking, and interference with emergency vehicles. We must collect and analyze traffic information to deal with these problems.

So far, many people have surveyed and analyzed vehicle information obtained by fixed cameras or sensors set up on the road. Such cameras or sensors can obtain only local information. Therefore, we must spend much time, money and labor to survey wide areas.

We believe that extraction of vehicles from aerial images is effective for solving the above problems. We can analyze a wide area at one time, as well as reduce the labor time and cost. Additionally, image processing has the possibility of automatic detection of new information.

Many methods of vehicle detection have been reported [1], [2]. However, these methods require the use of many sequential images or complex vehicle models, and hence require high computation cost.

In our study, we propose a method of extracting vehicles effectively from aerial images to collect and analyze traffic information in the city. We use the aerial image shown in table I.

II. ALGORITHM

We extract rectangular objects as the vehicles from the aerial images. Actual images are influenced by several objects, such as roadside trees, buildings, and white lines, and hence, we must decrease their influence.

The overview of our method is as follows.

- Extraction of white lines and road areas
- Extraction of edge pixels
- Extraction of center coordinates by voting process
- Extraction of vehicle frames by frame matching
- Refining of extracted vehicles

A. Extraction of white lines and road areas

We often use edge information in the automatic extraction of vehicles. If there are white lines near vehicles, their edges increase the possibility of false detection. We can reduce the rate of false detection by extracting white lines beforehand using the following features.

- high intensity
- low saturation
- narrow width
- pixel connection

When we detect edges in white line areas, we extract many edges that neighbor each other and are directed to the inside of the area (Fig. 1). White lines are extracted by the following processes.

(1) A pair of edge pixels satisfying the following conditions is searched.
- neighboring (8-connection)
- angle difference is π

TABLE I

AERIAL IMAGE

source	Tokyo Digital Map, Ltd.
equipment	ADS40
flight speed	about 300km/h
altitude	7266 feet
resolution	25cm
other	12000 pixels, Three Line Sensor

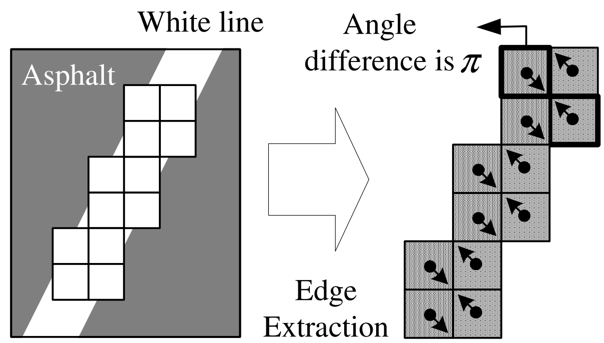

Fig. 1. Edge of white lines

K. Tanaka is with Graduate School of Informatics, Shizuoka University, Shizuoka, 432–8011, Japan gs06039@s.inf.shizuoka.ac.jp
H. Saji is with Graduate School of Science and Technology, Shizuoka University, Shizuoka, 432–8011, Japan saji@inf.shizuoka.ac.jp

Fig. 2. Input image

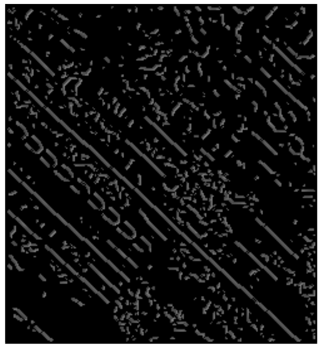

Fig. 3. Neighboring edge pixels

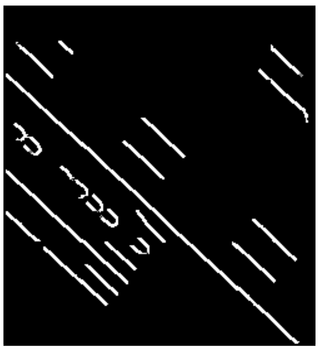

Fig. 4. White line areas

Fig. 5. Road areas

- directed to opposite directions and the inside of the area
- high intensity and low saturation

(2) Many connections are extracted as the white lines.

In the next step, we extract road areas by detecting gray areas that are adjacent to the extracted white lines.

The input image is shown in Fig. 2, and the neighboring edge pixel is shown in Fig. 3. The result of white line extraction is shown in Fig. 4 , and the result of road area extraction is shown in Fig. 5.

B. Extraction of Edge Pixels

We extract edge pixels that show the features of the objects. Because the following processes depend on the edge information, it is necessary to obtain high-quality edge information. These edges are extracted using Sobel operator. The edge intensity $I(x,y)$ and edge angle $\theta(x,y)$ are computed using Eq. 1 and Eq. 2. Then, in the following process, only edge pixels whose edge intensity values are greater than a threshold value are used.

$$I(x,y) = \sqrt{\delta x(x,y)^2 + \delta y(x,y)^2} \tag{1}$$

$$\theta(x,y) = \tan^{-1} \frac{\delta y(x,y)}{\delta x(x,y)} \tag{2}$$

Here, $\delta x(x,y)$ is the edge intensity in the direction of x, and $\delta y(x,y)$ is the edge intensity in the direction of y.

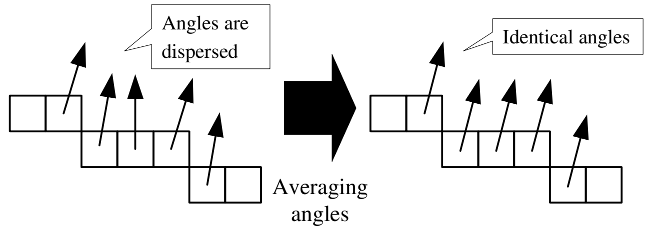

Fig. 6. Edge of white lines

Fig. 7. Features of white areas

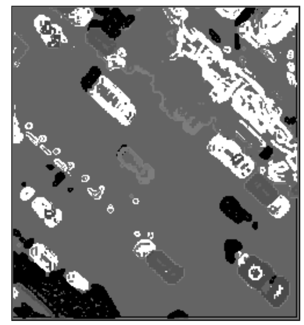

Fig. 8. Result of clustering

1) Averaging Edge Angles: Although Sobel operator can reduce noise and is used in many applications, the edge angles obtained with Eq. 2 are widely dispersed because of sampling errors, even when the edge pixels are located on the same straight line.

To solve the above problem, we average the edge angles. In the averaging method, we average the edge angle of 8-connection neighbors around the edge pixel and only the neighboring edge pixels that have small angular differences between the pixels. Therefore, the angles of edge pixels on the corners are not used. This process is shown in Fig. 6.

2) Clustering Edge Pixels: We use the edge clustering method. If many vehicles run side by side, the possibility of false detection increases. Therefore, we use a clustering method with pixel color information. In the clustering method, we use 14 levels of classification – black, white and high-saturation 12 levels. First, we classify the pixel into two levels by saturation. Next we classify the pixel of low saturation into two levels by intensity and the pixel of high saturation into 12 levels by hue information.

We explain concretely as follows. First, we extract black and white areas from the aerial image, and make a black area image and a white area image. The white area image is shown in Fig.7. Next, we extract edges from the saturation value image that is obtained by HSI transform. If the edge intensity is lower than a threshold, the edge is extracted from the black area image and the white area image. The edge having higher intensity value is selected from the two edges. If the edge intensity extracted from the saturation value image is higher than the threshold, the edge is extracted from the original aerial image and is classified into 12 levels by the hue information.

Therefore, the edge extraction and classification can help

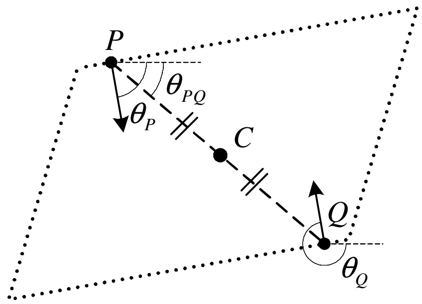

Fig. 9. Parameters used for extraction of center coordinates

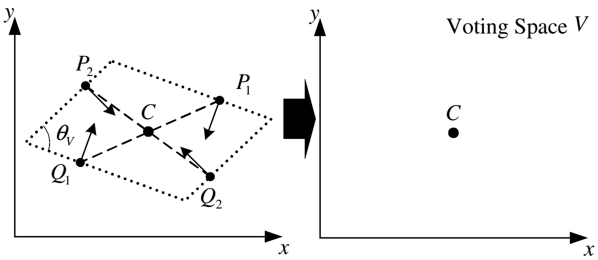

Fig. 10. Voting for center coordinates

to improve the following process efficiency. The clustering result is shown in Fig.8. The largest gray areas in the image do not have edge pixels.

C. Automatic Vehicle Detection

We propose a method of automatically extracting vehicles by the following processes.

(1) Extraction of center coordinates by voting process [3]
(2) Extraction of vehicle frames by frame matching
(3) Refining of extracted vehicles

1) Extraction of center coordinates by voting process: In this section, center coordinates are extracted by the voting process. Voting process is one of the process of Hough transform [4] that is a method of extracting straight lines by voting in parameter space and extracting peak values. In this study, we propose a method of extracting center coordinates of objects using both positional relations and the angle of the edge.

Parameters used for the extraction of center coordinates C are shown in Fig. 9. Here, P and Q are edge pixels, θ_P and θ_Q are edge angles of P and Q, and θ_{PQ} is the angle of line PQ.

To extract center coordinates, we use the geometric feature of center coordinates and a pair of edge pixels [5]. Edge pixels P and Q are symmetric about the candidates of center coordinates C and have opposite directions (meeting the condition of Eq. 3).

$$\pi - \alpha < |\theta_P - \theta_Q| < \pi + \alpha \tag{3}$$

If the edge pixels P, Q exist, C has the possibility of being the center coordinates.

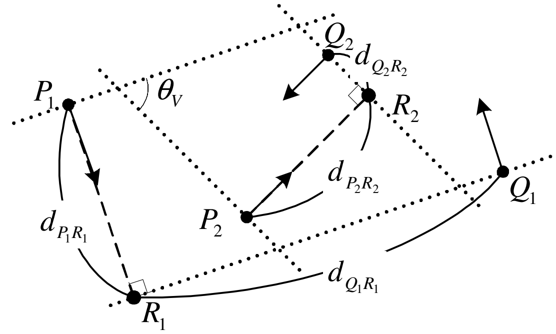

Fig. 11. Example of not Voting

On the basis of the above ideas, in order to reduce noise and extract center coordinates with high precision, we use two pairs of edge pixels. This process is shown in Fig. 10

In the voting process, the voting space V whose size is identical to the image space is used. Around candidates of center coordinates C, if two pairs of edge pixels that meet the condition of Eq. 3 and have an interval less than a threshold exist, and if vertex angle θ_V, which is composed of four edge pixels, is greater than a threshold value β (meeting the condition of Eq.4), we vote the coordinates C in the voting space V. After the voting process of these edge pixel pairs, the candidates of center coordinates are obtained by detecting coordinates whose numbers of votes are locally maximum in V.

$$\beta < \theta_V < \frac{\pi}{2} \tag{4}$$

Next, we describe the method of making this process more efficient. Applying this process may cause increases in the useless voting process and computation cost. Therefore, the following processes are applied.

(1) Use of clustering information of edges in section II-B.2
(2) Use of edge pairs with the inside direction (inward).
(3) Only when four edge pixels compose a quadrangle , the voting process is applied.

First, a pair of edges is selected on the basis of the clustering information. The use of only edge pairs that are close in class reduces the number of combinations, and this reduction increases the computation efficiency.

Secondly, only edge pairs with the inside direction are used. These edge directions meet the condition of Eq. 5.

$$|\theta_P - \theta_{PQ}| < \frac{\pi}{2} \tag{5}$$

Thirdly, an example of this process is shown in Fig. 11. Even if P_1Q_1 and P_2Q_2 meet the conditions of Eqs. 3 and 4, these edges do not compose a quadrangle. Therefore, if edges meet the conditions of Eq. 6, the voting process is applied to remove them.

$$\begin{aligned} d_{P_1R_1} &\geq d_{Q_2R_2} \times \sin\theta_V \\ d_{Q_1R_1} \times \sin\theta_V &\leq d_{P_2R_2} \end{aligned} \tag{6}$$

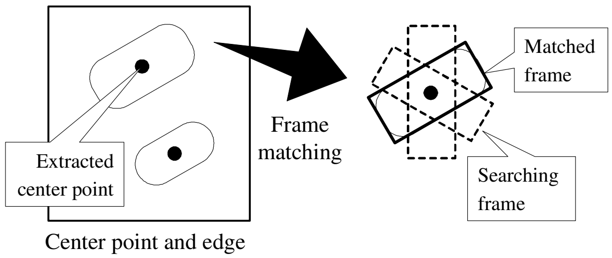

Fig. 12. Frame matching

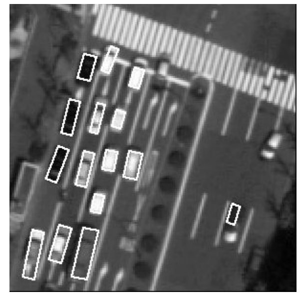

Fig. 13. Result 1

Fig. 14. Result 2

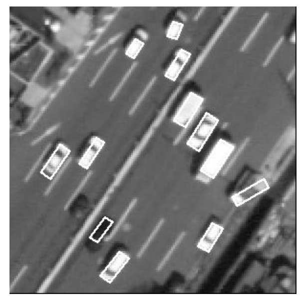

Fig. 15. Result 3

Fig. 16. Result 4

2) Extraction of vehicle frames by frame matching: In this section, we detect vehicle frames centered on the extracted center coordinates. This process is shown in Fig. 12. First, we prepare a template frame that has variable width, height and direction. Next, on all extracted center coordinates, matching scores of each frame are computed by varying the template frame position and size. Then, the frame that has the maximum score is selected as the vehicle frame. The matching score S_M is computed from :

$$\begin{aligned} \Delta\theta &= \theta_l - \theta(x,y) \\ S_M &= \sum s(x,y) \times \left(\frac{\pi^2 - (\Delta\theta - \pi/2)^2}{\pi^2}\right)^N. \end{aligned} \tag{7}$$

Here, $s(x,y)$ is the edge intensity in the image, $\theta(x,y)$ is the edge angle in the image, θ_l is the direction of the frame side, and N is a constant parameter. The angle difference $\Delta\theta$ is less than $\pi/2$. Equqation 7 indicates that the matching score becomes high when the edge angle in the image is equal to the angle of the side of the frame.

This method also involves a two-step approach for rapid processing. The first step determines a long side, and the second step determines a short side.

3) Refining of extracted vehicles: The processes described in sections II-C.1 and II-C.2 yield many candidates of vehicles. These candidates include incorrect frames. In this section, we delete incorrect frames. The refining process is executed by a four-step approach.

First, we delete a frame that includes wide road areas using data of road areas extracted in section II-A. Secondly, we delete a frame whose direction is different from the directions of the neighboring frames. Thirdly, we delete a frame whose aspect ratio is different from that of the general vehicles. Finally, we delete the frame that does not have the largest area among overlapping frames.

After the refining process, we obtain the final results of vehicle extraction.

III. EXPERIMENTAL RESULTS

We perform experiments using our method. The image size is 150 pixels×150 pixels. The results of automatic vehicle extraction by our method are shown from Fig. 13 to Fig. 16 and in Table II.

Before the experiments, the building areas were loosely removed by using a digital map [6], [7], and each parameter was selected at each image.

From Table II, extraction rate is seen to exceed 80% for all images, and there are markedly few false detections.

IV. SEMIAUTOMATIC AND MANUAL EXTRACTION

It is difficult to completely extract vehicles by our method. Therefore, we have made tools to extract vehicles manually or semiautomatically to solve the false detection and leak detection problems. The tool is shown in Fig. 17.

The semiautomatic detection method requires a center point, involves frame matching described in section II-C.2, and detects only one vehicle.

The manual detection method detects a vehicle on the basis of three vertexes. In the city, there are several shadow areas where vehicles are difficult to observe, and hence, we remove these shadow areas to facilitate manual detection.

A. Shadow Removal

It is difficult to detect vehicles in shadow areas from aerial images. We remove the shadow to enable clear manual detection of vehicles. The process of shadow removal is given below.

First, the sunny areas and the shadow areas are obtained. The shadow areas comprise connection pixel areas with low intensities. Next, average M_b and variance V_b in the sunny areas and average M_d and variance V_d in the shadow areas are obtained. The M_b and V_b values are target values in shadow removal. Finaly, the corrected C' is computed from C at the pixel in the shadow areas using Eq. 8.

$$C' = (C - M_d) \times \sqrt{\frac{V_b}{V_d}} + M_b \tag{8}$$

TABLE II

VEHICLE DETECTION RESULT

No	Total	extracted	not extracted	false	Extraction Rate
1	17	15	2	0	88.2%
2	8	7	1	0	87.5%
3	13	11	2	1	84.6%
4	10	8	2	0	80.0%
Total	48	41	7	1	85.4 %

Fig. 17. Manual extraction tool

Input images are shown in Figs. 18 and 20, and the results of shadow removal are shown in Figs. 19 and 21. After shadow removal, we can manually extract vehicles clearly.

V. CONCLUSIONS

In this study, we proposed a method of extracting vehicles effectively from aerial images, by using the voting process and frame matching. Our method can extract vehicle areas from images at a high extraction rate and can be applied to many applications. As future work in this study, we would like to perform experiments using several aerial images of various resolutions to obtain high-quality results.

VI. ACKNOWLEDGMENT

We would like to thank Japan Traffic Management Technology Association and Fire and Disaster Management Agency for their helpful support and comments on our study.

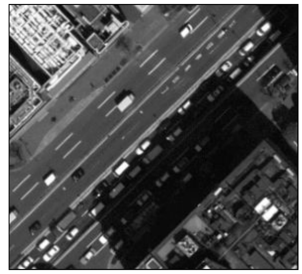

Fig. 18. Input image

Fig. 19. Shadow removal

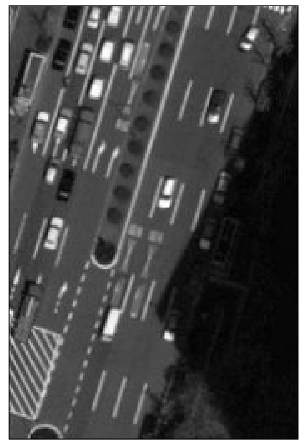

Fig. 20. Input image

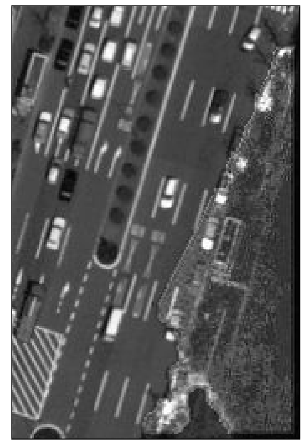

Fig. 21. Shadow removal

REFERENCES

[1] T. Fuse, E. Shimizu, and R. Maeda, "Development of techniques for vehicle maneuvers recognition with sequential images from high altitude platforms," *Journal of JSCE*, vol. 60, no. 737, pp. 159–173, 2003.

[2] S. Hinz and Munchen, "Detection of vehicles and vehicle queues in high resolution aerial images," *Photogrammetrie - Fernerkundung - Geoinformation*, pp. 201–213, 2004.

[3] K. Tanaka and H. Saji, "Detection of parallelograms using hough transform," *The Transactions of the Institute of Electronics, Information and Communication Engineers D*, vol. J89-D, no. 3, pp. 606–612, 2006.

[4] P. Hough, "Method and means for recognizing complex patterns," in *US Patent*, 1962.

[5] D. Ben-Tzvi and M. Sandl, "A combinatorial hough transform," *Pattern Recognition Letters*, vol. 11, pp. 167–174, 1990.

[6] K. Masuyama and H. Saji, "Extraction of vehicle areas from aerial images in the city," in *Journal of ViEW*, 2005, pp. 178–181.

[7] Y. Ogawa, S. Kakumoto, and K. Iwamura, "Extracting regional features from aerial images based on 3-d map matching," *The Transactions of the Institute of Electronics, Information and Communication Engineers D-II*, vol. J81-D-II, no. 6, pp. 1242–1250, 1998.